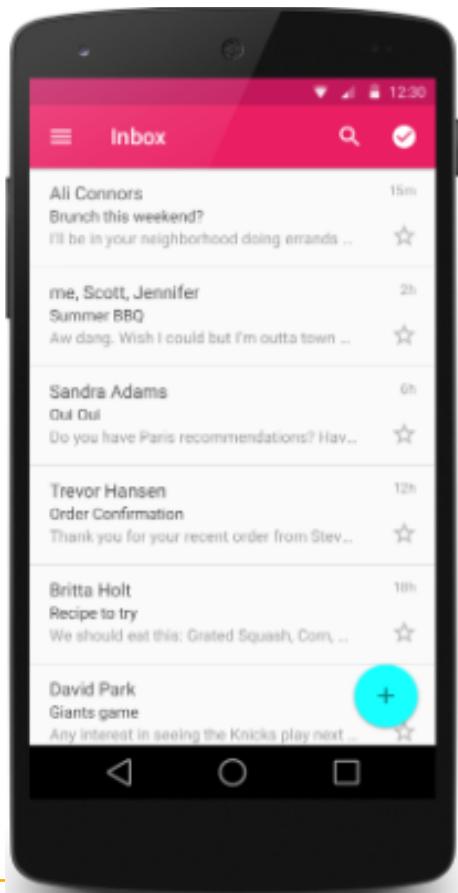


# Listas



A RecyclerView é uma versão mais avançada e flexível das listas.

Com ela podemos criar vários tipos diferentes de componentes que trabalharão juntos para a criação de listas personalizadas.

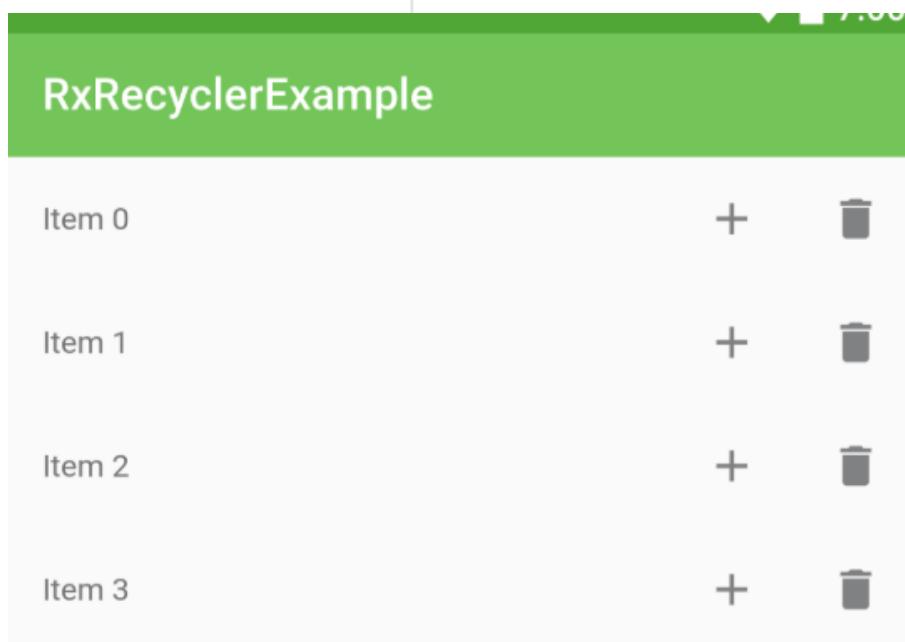
# Listas

```
1  <?xml version="1.0" encoding="utf-8"?>
2  <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
3      android:layout_width="match_parent"
4      android:layout_height="match_parent">
5
6      <android.support.v7.widget.RecyclerView
7          android:id="@+id/rv_contatos"
8          android:layout_width="match_parent"
9          android:layout_height="match_parent" />
10
11  </LinearLayout>
```



## Hens da Lista

```
1  <?xml version="1.0" encoding="utf-8"?>
2  <RelativeLayout
3      xmlns:tools="http://schemas.android.com/tools"
4      ...
5
6      <TextView
7          android:id="@+id/main_line_title"
8          tools:text="Line 1"
9          ...
10
11     <ImageButton
12         android:id="@+id/main_line_more"
13         android:src="@drawable/ic_add_black_24dp"
14         ...
15
16     <ImageButton
17         android:id="@+id/main_line_delete"
18         android:src="@drawable/ic_delete_black_24dp"
19         ...
20
```



# Adapter

Para alimentar os dados na sua lista, você precisará criar uma classe que estenda da RecyclerView.Adapter.

Este objeto irá criar visualizações para os itens que você definiu para a sua lista e irá te ajudar a popular eles com o conteúdo.

```
class MyAdapter(private val myDataset: Array<String>) :  
    RecyclerView.Adapter<MyAdapter.MyViewHolder>() {  
  
    // Provide a reference to the views for each data item  
    // Complex data items may need more than one view per item, and  
    // you provide access to all the views for a data item in a view holder.  
    // Each data item is just a string in this case that is shown in a TextView.  
    class MyViewHolder(val textView: TextView) : RecyclerView.ViewHolder(textView)
```



# Adapter

Ao  
recl  
Rec  
view  
dac

```
class ContatoViewHolder(itemView : View) : RecyclerView.ViewHolder(itemView) {  
  
    fun bindView(  
        contato: Contato,  
        clickListener: (Contato, Int) -> Unit,  
        position: Int  
    ) {  
        itemView.lbl_nome.text = contato.nome  
        itemView.lbl_conta.text = itemView.context.resources.getString(R.string.agencia_conta, contato.agencia, contato.conta)  
        itemView.lbl_banco.text = contato.banco  
        itemView.setOnClickListener({  
            clickListener(contato, position)  
        })  
    }  
}
```



# Listas

```
override fun onCreateViewHolder(parent: ViewGroup, viewType: Int): ContatoViewHolder {  
    val view = LayoutInflater.from(context).inflate(R.layout.item_contato_list, parent, false)  
    return ContatoViewHolder(view)  
}  
  
override fun onBindViewHolder(holder: ContatoViewHolder, position: Int) {  
    val contato = list[position]  
    holder.bindView(contato, clickListener, position)  
}  
  
override fun getItemCount(): Int {  
    return list.size  
}
```

Adiciona estilo de item na lista

Pega os itens da lista e vai adicionando nos itens do xml um por um

Retorna o tamanho da lista

